

EXPRESS MAIL LABEL NO. EL863783632US
DATE MAILED: January 2, 2002

PATENT

INVENTORS: Marvin J. RICH
Ashutosh MISRA

5 **VHDL TECHNOLOGY LIBRARY METHOD FOR EFFICIENT CUSTOMIZATION OF
CHIP GATE DELAYS**

This invention was made with government support under subcontract B338307 under prime contract W-7405-ENG-48 awarded by the Department of Energy. The Government has certain rights in this invention.

Cross-Reference To Related Applications

The present patent application is related to co-pending and commonly owned 15 U.S. Patent Application No. XX/XXX,XXX, Attorney Docket No. POU920010165US1, entitled "Delay Correlation Analysis and Representation for VITAL Compliant VHDL Models", and U.S. Patent Application No. XX/XXX,XXX, Attorney Docket No. POU920010005US1, entitled "Size Reduction Techniques for VITAL Compliant VHDL Simulation Models", filed on even date with the present patent application, the 20 entire teachings of which being hereby incorporated by reference.

Background of the Invention

1. Field of the Invention

5 This invention generally relates to the field of VHDL modeling, and more particularly relates to a system and method for updating a VHDL Technology Library for efficient customization of chip gate delays.

2. Description of Related Art

10 As ASICs (Application Specific Integrated Circuits) have become more complex, emphasis on verification techniques have flourished to assure that a particular ASIC's functionality can be verified prior to manufacture. One of the efforts is the IEEE VITAL (VHDL Initiative Towards ASIC Libraries) standard that allows back annotation of timing data into a simulation model. Part of this standard also defines the methodology required to generate VITAL compliant models. The VITAL standard provides the capability of generating very sophisticated behaviours of circuit behavior, which incorporate time delays (as determined by other timing tools).

20 Usually models of this type have the most meaning at the gate level, where a model is synthesized into gates associated with a particular technology. The provider of the technology usually provides a set of VITAL compliant VHDL models

EXPRESS MAIL LABEL NO. EL863783632US

for the gates, such that a very detailed behavior of the ASIC can be simulated. An event driven simulator is usually utilized with VITAL compliant models. During model load time, the SDF (Standard Delay Format file) is also read in to initialize a set of VHDL (Very High Speed Integrated Circuit Hardware Design Language) generic variables with the delay values. A naming convention exists for mapping SDF delay constructs to VHDL generic delay variable names, which is the basis of how the delays are back annotated. Due to the detail of the modeling, this type of simulation is most useful for going after specific scenarios where other simulation environments may be less accurate (i.e., clock gating, test logic, asynchronous boundaries, array controls, etc.).

Current state of the art for utilizing VITAL compliant VHDL models for simulation imposes a large size penalty, when modeling current ASIC chips. This size penalty is a consequence of the ever-increasing gate densities of ASIC chips, which require more instantiations of VITAL compliant VHDL gate behaviors. In conjunction with the increased gate counts, the SDF that associates timing delays to gates also increases in size at the same rate. It is the combination of VHDL model size and SDF size that influences the ultimate size of the resulting simulation model.

Currently, the size of the SDF generated for current chip densities rivals the size of the actual chip simulation model due to the number of wire segments that require delay specifications. This causes the SDF delay back annotation step, at simulation time, to take an ever increasing amount of time due to the large amount

EXPRESS MAIL LABEL NO. EL863783632US

of I/O to read the SDF from disk, the CPU processing time to transverse the SDF to extract delay values, and the processing time to back annotate actual delay values into instances of logic gates.

5 To incorporate a chip file into a specific technology, the technology manufacturer provides a technology library that describes the topology of all gate structures for that target technology. Benefits resulting from any reductions in the SDF file may be limited in scope or completely irrelevant, if a unique copy of the technology library is required for each chip. Any method that can alleviate the processing time for back annotation while preserving multiple chip usage of any manufacturer's technology library would be desirable for event simulation with delays using ASICs at the current or future chip densities.

10

15 Therefore a need exists to overcome the problems with the prior art as discussed above, and particularly for a method of updating a VHDL Technology Library for efficient customization of chip gate delays.

Summary of the Invention

20 A method and system update a VHDL technology library to incorporate correlated delay values by reading the VHDL technology library, inserting a `tpd_super_rise_time` generic declaration and a `tpd_super_fall_time` generic declaration for every VHDL gate model in the VHDL technology library, initializing

other generic variables in every VHDL gate model in the VHDL technology library to an equation representing a correlation policy; and outputting an updated VHDL technology library. Then, the method and system bind correlated delay constants in a 3-dimensional variable data array structure to a VHDL technology library using a

5 VHDL package embedded with the correlation delay data.

Brief Description of the Drawings

10 FIG. 1 is a block diagram illustrating a VHDL modeling system in accordance with a preferred embodiment of the present invention.

15 FIG. 2 is a more detailed block diagram showing a program memory in the system of FIG. 1, according to a preferred embodiment of the present invention.

15 FIG. 3 is a more detailed block diagram of a data memory in the system of FIG. 1, according to a preferred embodiment of the present invention.

20 FIG. 4 is an operational flow diagram illustrating an exemplary operational sequence for the system of FIG. 1, according to a preferred embodiment of the present invention.

EXPRESS MAIL LABEL NO. EL863783632US

FIG. 5 is a block diagram illustrating the files and tools utilized to generate an exemplary SDF file in the system of FIG. 1, according to a preferred embodiment of the present invention.

5 FIG. 6 is an operational flow diagram illustrating an exemplary operational sequence for analyzing an SDF file by the system of FIG. 1, according to a preferred embodiment of the present invention.

10 FIG. 7 is a circuit-timing diagram illustrating delays associated with a VHDL AND2_LOW gate in a VHDL model.

FIG. 8 is a graph displaying the distribution of delay values for interconnection paths in a typical VHDL file.

15 FIG. 9 is a graph displaying the distribution of delay values for different logic gate power levels of a typical VHDL file.

20 FIG. 10 is a data block diagram illustrating a unique, 1x, SDF super generic data structure of the system of FIG.1, according to a preferred embodiment of the present invention.

EXPRESS MAIL LABEL NO. EL863783632US

FIG. 11 is a data block diagram illustrating mapping correlation delays to a VHDL file of the system of FIG.1, according to a preferred embodiment of the present invention.

5 FIG. 12 is a table illustrating exemplary decode values of a unique AND2_MED logic gate for the system of FIG.1, according to a preferred embodiment of the present invention.

10 FIG. 13 is a logic circuit diagram illustrating the correlation of arbitrary logic blocks for the system of FIG.1, according to a preferred embodiment of the present invention.

15 FIG. 14 is a data block diagram illustrating a unique, 2x, SDF super generic data structure of the system of FIG.1, according to a preferred embodiment of the present invention.

FIG. 15 is a logic circuit diagram showing exemplary VHDL logic structures with identical delay topologies.

20 FIG. 16 is a 3X format illustration and decode table illustrating exemplary set of rise times with both positive and negative delays of the system of FIG.1, according to a preferred embodiment of the present invention.

EXPRESS MAIL LABEL NO. EL863783632US

FIG. 17 is a signal-timing diagram illustrating the concept of negative delays for the system of FIG.1, according to a preferred embodiment of the present invention.

5 FIG. 18 is an operational flow diagram illustrating an exemplary operational sequence for combining a set of delay data, for a particular logic gate instance, into a single delay generic for the system of FIG. 1, according to a preferred embodiment of the present invention.

10 FIG. 19 is an operational flow diagram illustrating an exemplary operational sequence for encapsulating a set of delay data for the chip into an array data structure for the system of FIG. 1, according to a preferred embodiment of the present invention.

15 FIG. 20 is a functional block diagram of an exemplary VHDL generation tool in the system of FIG. 1, according to a preferred embodiment of the present invention.

20 FIG. 21 is a functional block diagram of an exemplary VHDL compiler in the system of FIG. 1, according to a preferred embodiment of the present invention.

FIG. 22 is a functional block diagram of an exemplary VHDL simulator in the system of FIG. 1, according to a preferred embodiment of the present invention.

Description Of The Preferred Embodiments

The present invention, according to a preferred embodiment, overcomes problems with the prior art by providing a unique process that unbinds the rise/fall tuples from the generic variable name, thereby enabling other methods to reduce the size of a VITAL compliant VHDL simulation model. The simulation model reduction is achieved through the significant reduction in size of the SDF file required to back annotate delay values into the model. The reduced size results in significantly reduced memory requirements for a computing system. This reduces costs of the overall computer system required for simulation. The use of this process also has the effect of increasing performance on the computer platform that is host to the simulation model, due to reduced memory paging requirements and reduced file I/O.

Exemplary embodiments disclosed are for a model written in VHDL, but similar techniques may also be applied to simulation models written in the Verilog language. The techniques involve editing the VHDL model or its associated SDF, based on size reduction observations, to realize a savings in the size of the resulting simulation model.

The SDF size reduction is based on the correlation of disparate delay values, whose scope in prior art was limited to a single instance of a logic gate. These correlated values will tend to cluster around technology dependent values, such that the same delays can be reused regardless of the chip size. This provides the

EXPRESS MAIL LABEL NO. EL863783632US

advantage that the SDF size reduction utilizing this technique, will scale well with increased chip size, resulting in a larger percentage size reduction for the larger, and more problematic, chip sizes. All improvements utilize techniques that maintain compliance to the VITAL standard. This has the advantage of allowing any simulation platform that already implements the VITAL standard to easily incorporate this mechanism into its technology library in order to gain the benefits of this invention.

FIGs. 1 and 2 illustrate an exemplary VHDL modeling system according to a preferred embodiment of the present invention. The VHDL modeling system 100 includes a computer system 110, having VHDL tools 114 and SDF tools 116. The computer system 110, according to the present example, includes a controller/processor 122, which processes instructions, performs calculations, and manages the flow of information through the computer system 110. Additionally, the controller/processor 122 is communicatively coupled with program memory 112. Included within program memory 112 are VHDL tools 114 and SDF tools 116 (which will be discussed in later in greater detail), operating system platform 118, and glue software 120. The VHDL tools 114 contain a VHDL generator 208, a VHDL correlation generator 210, a VHDL compiler 212, a VHDL simulator 214 and a VHDL Library Modifier 216. The SDF tools 116 consist of an SDF generation tool 202, an SDF analyzer 204, and an SDF reducer 206. The operating system platform 118 manages resources, such as the data stored in data memory 124, the scheduling of tasks, and processes the operation of the VHDL tools 114 and the SDF tools 116 in

EXPRESS MAIL LABEL NO. EL863783632US

the program memory 112. The operating system platform 118 also manages a graphical display interface (not shown), a user input interface (not shown) that receives inputs from the keyboard 106 and the mouse 108, and communication network interfaces (not shown) for communicating with a network link (not shown).

5 Additionally, the operating system platform 118 also manages many other basic tasks of the computer system 110 in a manner well known to those of ordinary skill in the art.

10 Glue software 120 may include drivers, stacks, and low level application programming interfaces (API's) and provides basic functional components for use by the operating system platform 118 and by compatible applications that run on the operating system platform 118 for managing communications with resources and processes in the computing system 110.

15 Each computer system 110 may include, *inter alia*, one or more computers and at least a computer readable medium 128. The computers preferably include means 126 for reading and/or writing to the computer readable medium 128. The computer readable medium 128 allows a computer system 110 to read data, instructions, messages or message packets, and other computer readable information from the computer readable medium. The computer readable medium, 20 for example, may include non-volatile memory, such as Floppy, ROM, Flash memory, disk drive memory, CD-ROM, and other permanent storage. It is useful, for

EXPRESS MAIL LABEL NO. EL863783632US

example, for transporting information, such as data and computer instructions, between computer systems.

FIG. 3 illustrates a preferred embodiment of the data memory **124** of the VHDL modeling system **100** of FIG. 1. Data memory **124** includes a wire delay file **302**, a synthesized chip netlist **304**, a technology library **306** and technology rules **308**. The wire delay file **302** is usually derived after physical gate layout and contains the physical wire delay information (rise and fall RC time constants and load capacitance) that influences external gate delays (e.g. gate connection). The synthesized chip netlist **304** is the resulting file after the logical representation of a chip has been synthesized into an equivalent gate level representation for a target technology. A target technology represents a chip manufacturing process that provides a defined chip density and performance level (e.g. CMOS7S .18 micron technology from IBM Corporation). The technology library **306** is a file, independent of the chip, which describes the topology (e.g. number of inputs, names, number of outputs, etc.) of each gate for a target technology. The technology rules **308**, also independent of the chip, contain detailed delay information associated with the internal operation of each logic gate.

The data memory **124** also contains an SDF (Standard Delay Format) file **310**, an SDF analysis file **312**, and a reduced SDF file **314**. The SDF (Standard Delay Format) file **310** is an industry standard (IEEE 1076.4) file that specifies delays in a format for simulation tools to back annotate timing delays and is created

EXPRESS MAIL LABEL NO. EL863783632US

by the SDF generation tool 202. The SDF analysis file 312 is output by the SDF analyzer 204 and can be used to determine delays used for each VHDL generic or to determine delays associated with each instance of a logic gate. The reduced SDF file 314 is a file output by the SDF reducer 206 and contains only two generics per instance of each logic gate.

Also contained in data memory 124 are a chip VHDL file 316, a correlation VHDL file 318, and an object file 320. The chip VHDL file 316 is a file output by the VHDL generator 208 for use by a VHDL event simulator 214. The correlation VHDL file 318 has correlation delay information embedded within the file, and the object file 320 contains the machine language executables used to simulate the chip on a particular workstation platform (e.g. Windows, Unix, etc.) The data memory 124 may optionally contain files such as a log file 322 and a checkpoint file 324 to save the current state of a simulation.

FIG. 4 is an exemplary operational flow diagram illustrating the overall process of preparing a chip netlist for VHDL simulation for the system of FIG. 1. The system enters the process, at step 401, where the decision is made as to which path to execute. Path A performs the steps necessary to analyze and correlate gate delays to realize the SDF size reductions as specified in this invention. Path B performs a traditional chip VHDL compile, using prior art methods. Path C involves the alteration of the technology library to accommodate a reduced SDF delay specification format as specified in this invention. Each of these paths can be

EXPRESS MAIL LABEL NO. EL863783632US

executed independently, until all paths converge, at step **420**. Paths A and B are executed for each chip simulation iteration. Path C is executed only once for the target technology in which the chip will be manufactured.

5 If path “A” is chosen, the process proceeds, at step **402**, to create an SDF file **310** for a given synthesized chip netlist **304**. The processing, at step **402**, involves prior art methods as specified in the IEEE VITAL specification. The SDF file **310**, at step **404**, is analyzed by the SDF analyzer **204**, resulting in an SDF analysis file **312**, according to methods to be described later in greater detail. The resulting SDF analysis file **312** is used, at steps **406**, **408**, to perform an SDF reduction and generate a correlation VHDL file **318**, according to methods to be described later in greater detail. The correlation VHDL file **318** is compiled, at step **410**, utilizing a VHDL compiler **212**.

10 If Path B is chosen, the process proceeds, at step **412**, to generate a chip VHDL file **316** and then compile, at step **418**. The VHDL generation, at step **412**, utilizes a VHDL generation package **208**, which takes a synthesized chip netlist **304** and generates technology specific VHDL, as illustrated in FIG. 20. The VHDL generation tool **208** is usually bundled with a synthesis tool. The VHDL compile, at step **418**, utilizes an existing VHDL compiler **212**, which takes IEEE compliant VHDL as input, and generates VHDL object code suitable to be utilized by an associated VHDL simulator **214**.

EXPRESS MAIL LABEL NO. EL863783632US

If Path C is chosen, the technology library **414** is updated one time, at step **414**, and compiled, at step **416**. The single update, at step **414**, is normally all that is required because the target technology library **414** is usually constant at the gate level for a particular chip, or family of chips. The update of the technology library **414** is done in order to provide a mechanism for binding chip specific delay information in a generic fashion, such that a single technology library **414** can still be utilized for multiple chips. The updated VHDL is compiled, at step **416**, using a VHDL compiler **212**.

At step **420**, a VHDL simulation is executed. All paths (A-C) must be complete at this step, such that the SDF and compiled VHDL files are available for simulation as depicted in FIG. 22. Any simulator that supports IEEE compliant VHDL format is suitable for this step.

FIG. 5 is a block diagram illustrating the files and tools utilized to generate an exemplary SDF file **310** in the system of FIG. 1, according to a preferred embodiment of the present invention. The SDF generation tool **202** may be a custom or vendor provided tool that takes the synthesized chip netlist **304**, wire delay information **302**, and the technology parameters (the technology library **306** and technology rules **308**) as input to generate an SDF file **310**.

FIG. 6 is an operational flow diagram illustrating an exemplary operational sequence for analyzing an SDF file **310** by the system of FIG. 1, according to a

EXPRESS MAIL LABEL NO. EL863783632US

preferred embodiment of the present invention. The intent of this sequence is to take as input an existing SDF file **310** generated as shown in FIG. 5. The SDF file **310** is utilized to perform delay correlation analysis. Once the correlation process completes, the correlation results are placed in an SDF analysis file **612**, which can 5 be utilized by multiple downstream processes to exploit the correlation savings.

The correlation process produces a reduced set of delays, by performing delay correlation analysis across an entire chip as represented in the SDF file **310**. The correlation process shown correlates delays according to a policy of combining common delays for a common delay generic name. It will be obvious to those of ordinary skill in the art, in view of the present discussion, that alternative embodiments of the invention are not limited to this correlation policy (a set of criteria used to combine delays). Other policies could be utilized to take advantage of common delay properties for a chip, for example, that can be exploited if delay specifications are not bounded by the scope of a single logic gate instance, as 10 15 discussed in the examples provided herein.

The SDF analyzer **204** enters the sequence, at step **602**, where it collects all the unique generic variable names specified in an SDF file **310**. For example, **tpd_A** 20 **702** (see FIG. 7) is a generic name (delay name) for the propagation delay from a previous gate to pin A of a logic gate. The SDF file **310** contains values for all the delays for every gate in the synthesized chip netlist **304**. So, for example, for the gate shown in FIG. 7, the **delay1** **710** is the delay from the driving gate to this pin

EXPRESS MAIL LABEL NO. EL863783632US

(e.g. `tpd_B 704`); `delay2 712` is the internal delay from the gate I/O pad to the internal AND circuit (e.g. `tipd_A`); and `delay3 714` is the delay to the output pin Z due to a change in pin A or pin B input pins (e.g. `tpd_A_Z` and `tpd_B_Z 708`). Each usage of this particular `AND2_LOW` gate **700** would have a unique set of delays associated with it (rise and fall times).

5

A particular generic, such as `tpd_A`, is selected, at step **604**. Then, at step **606**, all the delays assigned to this generic for the entire chip are extracted from the values in the SDF file **310**. A list of logic gate instances that reference this generic is maintained in data memory and placed in the SDF analysis file **612** on completion. For example, both `gate1: AND2_LOW` and `gate2: AND2_LOW` are instances of the `AND2_LOW` usage such that separate indexes would be needed for the `tpd_A` generic. The delays are preferably sorted in ascending order, at step **607**, and any duplicate delay entries are removed. At step **608**, the sorted delays are grouped into sets of up to 62 entries corresponding to correlation sets (this is explained later in greater detail). At step **610**, the SDF analyzer **204** determines if all the generics of the SDF file **310** have been analyzed. The process, beginning at step **604**, is repeated for each generic of the SDF file **310**. At step **612**, an SDF analysis file **312** is generated. The SDF analysis file **312** contains delay data for the entire chip, which has been correlated according to a particular correlation policy. This file encapsulates the inherent delay redundancies across the entire chip for a particular policy. An example of a correlation policy would be the correlation of delays with same generic name. The SDF analysis file **312** is utilized, at step **406**, to generate a

EXPRESS MAIL LABEL NO. EL863783632US

reduced size SDF file 314 (no explicit delays specified), and to bind the technology library 306 to the set correlated delays, at step 408. A system implementation could also use this file to generate statistics for a particular correlation policy, such that a plurality of unique correlation policy results could be compared for highest efficiency.

5

In practice, careful observation has shown that delays are not strictly uniformly distributed. Synchronous logic has a bounded cycle time requirement, where the latch-to-latch propagation delay, through combinatorial logic, must be less than the chip cycle time. Therefore, delays are usually clustered about a range of values, with an upper bound delay equal to the cycle time. In order to allow for transversal of multiple combinatorial logic gates between latches, the majority of the delays are clustered around an even smaller range of values, relative to the cycle time. As shown in FIG. 8, interconnect delays will cluster around certain points dependant upon path lengths. Short delays 802 are for short path lengths and longer delays 804 are for long path lengths. Although not indicated in the figure for purposes of clarity, there is a possibility of overlap of the sets of values. For intra-circuit delays, values will cluster around the drive capabilities (such as speed, power). FIG. 9 indicates the distribution of logic gate delays according to power levels (high 902, medium 904, and low power 906). Again, though not shown in this example, there exists the possibility of overlap.

EXPRESS MAIL LABEL NO. EL863783632US

Assuming the binding of rise/fall tuples to VHDL generics is broken, coupled with the observation that delay values tend to cluster, there could be a very narrow range of delay values. For example, in the VHDL generic:

tpd_A = (rise time, fall time),

5 if the rise time were a separate object from fall time, now the worst-case probability for a delay match, in the range 0 ns to .999 ns in 1 ps intervals, would be 1 in one thousand. If clustering occurs due to the technology, the probability of a match could be in the range of 1 in one hundred. This decoupling of tuples provides a mechanism for recognizing redundancies across logic gates, which opens up the set 10 of delays that can exploit this technique.

FIG. 10 illustrates a unique data structure **1000** that captures the net effect of the delay correlations without having to specify the same amount of data as in a traditional SDF file **310**, but still maintains SDF compatibility to the VITAL specification. This data structure **1000** will be referred to as a 1x data structure 15 hereafter, for reasons that will soon become apparent. A great advantage of this 1x data structure is that existing software can readily utilize this technique. The 1x data structure **tpd_super 1000** represents a “super generic” value in a reduced SDF file **314** that encapsulates *all* of the delay values for a particular gate. There will be, at 20 most, only two generic specifications in the reduced SDF file **314**, one rise time generic variable for rise time values and one fall time generic variable for fall time values. Generic values that are one dimensional, such as clock pulse width, may be specified in either or both the rise time and fall time generics. The first position **1002**

EXPRESS MAIL LABEL NO. EL863783632US

of the 1x data structure contains an index value for the correlation set. The value of the index is represented by one of 62 characters: the numerals 0 to 9, the lower case alphabet a to z, and the upper case alphabet A to Z. The remaining positions of the 1x data structure represent actual delay values in the correlation set for specific delays in the generic.

For example, as shown in FIG. 10, the 1x data structure representing the rise times for an AND2_LOW gate is a string of 7 digits; the “0” position representing the index position in a correlation set, the “1” position representing the tpd_A delay value position, the “2” position representing the tpd_B delay value position, the “3” position representing the tipd_A position, etc.

It is worthwhile to note that the set of valid characters allowed by VHDL are those defined by the ISO 8859-1 standard, which defines 256 (2^8) characters. Therefore, the maximum number of delays that could be allowed is 256. However, aside from the 62 alphanumeric characters, the other characters are either messy (\$%*@) or unprintable. So, the number 62 was chosen for illustrative purposes, whereas, the maximum number of delays in this example could actually be up to 256.

A separate exemplary data structure 1104, shown in FIG. 11, and that will be utilized in the correlation VHDL 408, contains the constants (actual delay values) that equal the delays for all of the gates. This data structure 1104 comprises an

EXPRESS MAIL LABEL NO. EL863783632US

array (may be also referred to as a matrix) of delay values preferably organized for efficiently indexing and retrieving the values from the data structure **1104**. This structure **1104** is also used to bind the correlated delay values to the VHDL technology library **306** via a VHDL package (a VHDL construct that allows for sharing of common data items). A 3-dimensional variable array structure is utilized to most efficiently specify correlated delays. The z-axis **1106** of the data structure represents a set of common blocks for each logical topology (e.g AND2_LOW (low power), AND2_MED (medium power), AND2_HIGH (high power) are one set of common blocks: AND2_NEW). Each entry on this axis depicts logic gates with a common topology (same amount and type of delays). On the x-axis **1108** each position represents a delay value for the gate topology (e.g. slot 1 = tpd_A for AND2_NEW). The y-axis **1110** contains the actual delays.

The 3D variable array structure **1104** is used for efficient representation of correlation delays for a particular correlation set. The X-axis and Y-axis depths are not necessarily equal to each other, nor are they necessarily equal to the X or Y-axis depths for another Z entry. Expressing the variable dimensions of the 3D array structure **1104** as a set of 3-tuples, where each variable dimension would be an element {Z, X, Y} in the set where:

- Z = Logic topology type selector
- X = Maximum Generic Selection Slot
- Y = Maximum Number of Correlation Entries

EXPRESS MAIL LABEL NO. EL863783632US

a typical tuple $X_1Y_1Z_1$ for a two input AND gate would be $\{5,6,40\}$, where 5 is the entry for a two input AND gate topology, and 6 is the number of generics, and 40 would indicate a maximum of 40 unique delay values in the correlation set. The X_1 value of 6 is defined by the gate topology, the Z_1 and Y_1 values would vary, based on the processing order (Z_1) and delay correlation (Y_1). A larger gate (e.g. Latch) topology would have a different delay correlation capacity identified by $Z_2X_2Y_2$ tuple $\{2,20,60\}$ (Logic gate topology 2, Total of 20 generics defined, Total of 60 unique delay values utilized in this correlation set). The following calculations demonstrate the efficiencies of using this variable structure approach versus a uniform dimension array *for just two entries*:

Given: 4 bytes per slot entry

$$\begin{aligned} \text{Entry 1: dimension 3-tuple} &= \{5,6,40\} \\ \text{Entry 2: dimension 3-tuple} &= \{2,20,60\} \\ X_{\max} &= \max(X_1, X_2) = \max(6,20) = 20 \\ Y_{\max} &= \max(Y_1, Y_2) = \max(40,60) = 60 \end{aligned}$$

Calculations:

Uniformed Array Size Requirement (2 entries)

$$\begin{aligned} 2 \text{ entries} * (X_{\max} * Y_{\max}) \text{ slots/entry} * 4 \text{ bytes/slot} &= (2)*(20 * 60)*4 \\ &= 9,600 \text{ bytes} \end{aligned}$$

Variable Array Structure Size Requirements (2 entries):

Entry 1:

$$(X_1 * Y_1) \text{ slots} * 4/\text{bytes/slot} = (6) * (40) * 4 = 960 \text{ bytes}$$

Entry 2:

$$(X_2 * Y_2) \text{ slots} * 4 \text{ bytes/slot} = (20) * (60) * 4 = 4800 \text{ bytes}$$

EXPRESS MAIL LABEL NO. EL863783632US

Total Variable Array Size = 4800 + 960 = 5760 bytes.

Savings using variable array (two entries) = 9600 - 5760 = 3,840 bytes

As the number of 3D entries for a chip will normally run in the hundreds, the savings realized can be extrapolated.

5

As an example, suppose the correlation profile of AND2_LOW, AND2_MED, and AND2_HIGH are disjoint and resemble the distributions shown in FIG. 9. To represent the delays for all three types of AND2_xxx gates in a single generic would look like:

10

AND2_LOW:

tpd_super_rise = "0aQ3478"

tpd_super_fall = "0tu8AcT"

AND2_MED:

tpd_super_rise = "1y76Q25"

tpd_super_fall = "14Xampl"

AND2_HIGH:

tpd_super_rise = "2uvwxyz"

tpd_super_fall = "2abcdef"

15

20 The decoding of the super generic of AND2_MED where tpd_super_fall = "14Xampl", is illustrated in FIG. 12. The character value of position 0 shows that the position of the correlation set for the delays of AND2_MED is 1. The delay value for position 1, tpd_A="4", is located at the 5th position in the correlation set.

EXPRESS MAIL LABEL NO. EL863783632US

This procedure can be taken a step further in order to correlate arbitrary logic blocks, such as those shown in FIG. 13, using the Z entry (which is the logic gate topology index). Gates with similar topology (AND2, OR2...) would have a common Z entry. One can correlate across different topologies when the Z entry becomes 5 “the generic entry” (e.g. tpd_A). Now any logic gate that uses tpd_A knows to utilize that entry. VHDL allows for associative array indexes (or pointers), such that the technology library 306 would literally use an index (or pointer) of “tpd_A”, which would map to the proper Z entry. This allows for correlations across a wide range of logic gates, realizing significant savings. The gate structure AO22 1304 is actually a 10 combination of two AND2 gates 1302 (previously discussed) and an OR2 gate. This correlation has a much larger set of delays with a higher probability for correlating 15 the data.

In order to optimize delays across functional gates usually requires more 15 capacity than the 1x data structure 1000 can provide. This method likely requires a 2x data structure 1400 (shown in FIG. 14), where the first set 1402, preferably, indexes to 1 of 62 possible slots and the second structure 1404, preferably, indexes 20 to 1 of 62 possible delay values in that slot. The 2x data structure 1400 is so named because it is approximately twice the size of the original 1x data structure 1000 (e.g. for the AND2 gate, there are 12 characters vs. 7 characters).

A correlation can be performed on a pin type such as tpd_A. Then, any delays to gates that have a tpd_A pin will be correlated as a group. Now, delays

EXPRESS MAIL LABEL NO. EL863783632US

from gates with different functional operations, such as those shown in FIG. 15 can be merged. All of the gates in FIG. 15, and possibly others, would have a commonly named tpd_A and tpd_B pin.

5 When correlations are performed within a gate topology, the actual VHDL model that would incorporate the delay values knows ahead of time that AND2, OR2, etc. have a 6-entry 2x format and can pick off the right values in the data structure. If the correlation were performed across gate topologies, the actual VHDL model would still know which generic entries to select. Therefore there are minimal
10 VHDL updates to the existing VHDL. A one time conversion of the technology library **306** assures that the AND2_xxx VHDL models, for instance, point to a fixed location where tpd_A data would reside.

15 Taken one step further, a 3x data structure **1600** (shown in FIG. 16) can represent 238,328 (62^3) possible delays. This allows *all* the delays to be uniquely specified, without any correlation, using only data structure conversions and decodes. The tradeoff is that there are more characters in the reduced SDF file **314**, but there would still be a significant SDF reduction that results in size and time savings. The 3X data structure **1600** typically pays a flat storage penalty up front. It
20 is typically larger by 50% over 2X structure per gate. It is still a very efficient representation versus a conventional SDF, yet it can accommodate the range of gate delays for most chips. The 1X and 2X structures pay a smaller price, but also

EXPRESS MAIL LABEL NO. EL863783632US

supply a correlation array structure. If the delay correlations are good, the 1X or 2X structures will be more efficient overall (total simulation model size).

The 3x data structure **1600** handles both positive and negative delays, as shown in FIG. 17. So, there will be an efficient representation of negative delays also. A key observation of this 3x data structure **1600** is that the range of negative delays is usually much smaller than positive delays because negative delays involve an “overlap” case that is valid for a short period of time relative to a reference point.

The asymmetry in the range of positive vs. negative delays can be utilized in order to keep the structure indexes minimal by not requiring a sign for each delay. A negative “base delay”, such as -0.500ns, is assigned to the 0 position, and the value of each position is increased by 1ps per position for 238,327 increments. So, the entire range of -0.500ns to +237.327ns can be realized by a single 3x data structure **1600**, where the delay is basically a base 62 number plus a signed offset base number.

In actuality, this method allows for a 3X range of $(2^8)^3 = 2^{24}$ or 16 million increments, which amounts to a 16 microsecond range at 1 picosecond intervals. This is more than enough of a range of delays to accommodate any chip. However, the 3X structure is a scalable format, where 4X, 5X.... nX structures could be easily constructed and utilized with potential size savings. Anything beyond a 3X structure should not be needed for most chips.

EXPRESS MAIL LABEL NO. EL863783632US

The VHDL model would have fixed equations such as:

$$\text{Delay} = \text{Base} + (\text{tpd_A_1} * 62^2) + (\text{tpd_A_2} * 62^1) + (\text{tpd_A_3})$$

5 where, `tpd_A_1`, `tpd_A_2`, and `tpd_A_3` are the decoded character values (0-61) of the first position in each of the three sets of the 3x data structure **1600**. The 3x data structure **1600** in FIG. 16 indicates values of `tpd_A_1` = 1 (1), `tpd_A_2` = 10 (a), and `tpd_A_3` = 52 (Q). So, in this case:

10
$$\text{Delay} = -500 \text{ ps} + (1 * 62^2) + (10 * 62) + 52$$

$$\text{Delay} = -500 + 3844 + 620 + 52 \text{ ps} = 4016 \text{ ps} = 4.016 \text{ ns}$$

Note that the equation is evaluated in ps in order to work with whole numbers.

15 Usually the delays associated with timed synchronous logic are bunched around a particular cycle time. However, for severely disjoint ranges, the 2X structure should be used in order to decrease the overall range of delay values. Also, the SDF analysis file **312** allows for certain portions of the SDF to remain untouched, for robustness.

20 After an SDF analysis file **312** has been created, the SDF reducer **206**, as shown in the operational flow diagram of FIG. 18, uses it to create a reduced SDF file **314** that is much smaller in size. The reduced SDF file **314** is still a VITAL

EXPRESS MAIL LABEL NO. EL863783632US

compliant SDF with a significantly reduced number of generics (two generics per logic gate instance). The SDF file **310** is built on a per instance basis and each instance contains tuples of the rise and fall times of each delay in the structure. The

SDF reducer **206** enters the process, at step **1802**, and selects an instance of a gate

5 (e.g. gate1: AND2; gate2: AND2 are two instances of the AND2 gate with separate

delay values) from the SDF file **310**. At step **1804**, all the delay values for the

selected instance are collected from the SDF analysis file **312**. Then, at step **1806**,

the SDF reducer **206** builds the two single super generics `tpd_super_rise` (rise

times) and `tpd_super_fall` (fall times) for the selected instance. It is understood that

each super generic will be represented by a collection of pointers into a data

structure array (or matrix) containing all the relevant delay values. For each

instance, in the SDF file **314**, the collection of pointers, that point into the data

structure array (or matrix) for the super generic to be able to resolve the actual delay

values for the particular instance, takes up significantly much less storage than a set

5 of conventional generics (storing information to conventionally identify actual delay

values) for a similar instance. Advantageously, the size of the pointers for each

instance, according to the preferred embodiment of the present invention, will

typically be significantly smaller (more efficient) in overall storage requirements than

the storage requirements of the information stored for delay values associated with

20 instances in a conventional implementation SDF file. This novel process is repeated

for every instance of every gate. Therefore, the storage efficiencies are multiplied by

the number of instances in an overall SDF file. If all the instances, at step **1808**,

have been converted, then a significantly reduced SDF file **314** is output, at step

EXPRESS MAIL LABEL NO. EL863783632US

1810. Besides the increased storage efficiency by removing duplicate delay values and storing these in an organized fashion in the super generics, the use of the memory efficient pointers in an SDF file will additionally reduce the amount of memory storage used for an implementation. This is an important advantage of the present invention over any known prior art systems.

5

The process of generating the correlation VHDL file **318**, according to step **408**, is shown in FIG. 19. The VHDL correlation generator **210** enters the process at step **1902** where the correlation delays are extracted from the SDF analysis file **312**. The VHDL correlation generator **210**, at step **1904**, generates a VHDL associative array structure (or matrix structure) such that, for example, character "a" is used as an array index (or index into the matrix). Then, the correlation VHDL file **318**, which is a unique VHDL package file with correlation data embedded, is output, at step **1906**.

10
15

The overall process of generating a VHDL file **316**, according to step **412**, is shown in FIG. 20. The VHDL generator **208** uses the synthesized chip netlist **304** and the technology library **306** to create a chip VHDL file **316**. The chip VHDL file **316** is the representation of the chip in the VHDL language.

20

The technology library **306**, which is VHDL code describing the behavior of the logic gates, is only updated once, at step **414**, and is independent of the actual delays. It can be done prior to building a simulation model for a chip. This allows

EXPRESS MAIL LABEL NO. EL863783632US

binding the delays in the VHDL gate description to a specific chip delay profile without requiring unique copies of the Technology Library **306**. This one time update of the Technology Library VHDL **306**, is based on pre-determined gate topologies. The generic value positions in the structure are known ahead of time, and the actual entries are from the `tpd_super_xxx` generics for the gate.

5

Current state of the art provides a mechanism to back annotate delays using a VITAL compliant SDF file. The concept of correlating decoupled rise and fall delays, and exploiting this correlation with reduced SDF structures, provides the potential for a much smaller and more efficient event simulation model with delays. To exploit these potential savings, the correlated delay values (1X or 2X formats) must still be communicated to the Technology Library VHDL models **306**, which now only have two generics specified. A unique mechanism provides an efficient VHDL compliant mechanism to automatically customize a Technology Library **306** at simulation time with actual delay values, using a condensed set of correlated delays, by providing a unique interface to access correlated delays, that eliminates explicit back annotation of delay values. This discussion discloses a preferred embodiment of an interface to exploit the 1X and 2X structures respectively.

10

15

Given the 3D correlation structure, the Z-axis entries for 1X structures are typically logic gates with similar topologies that have correlated sets of delays. The set of Z-axis entries could be thought of as a set of arrays Z_1, Z_2, \dots, Z_n where each array is two dimensional, such that array entry Z_n be viewed as an array $Z_n(X_n, Y_n)$,

EXPRESS MAIL LABEL NO. EL863783632US

where Z_n is the nth entry in data structure representing a set of delays associated with a common correlation policy. The dimension X_n represents sets of delay generics, and dimension Y_n are the correlated delay values. For 1X structures, a delay correlation policy would typically be across gates with a common topology.

5 For 2X structures, a typical correlation policy would be delays associated with a common generic name.

10 The set of arrays $Z_1 - Z_n$ would be defined as a set of VHDL compliant array constants (output of 408), which are compiled into a VHDL package body (output of 410). VHDL semantics allow independent compilation of VHDL constant declarations and actual values, such that binding can be deferred until simulation time. This is also known as late binding at run time. Utilizing this capability, a unique mechanism can be derived that automatically maps the delays encapsulated by the 1X and 2X data structure to a Technology Library VHDL model 306 with no 15 explicit delay back annotation.

20 The VHDL Technology Library modifier 216 updates the Technology Library 306 by inserting the tpd_super_rise and tpd_super_fall generic declarations for every VHDL gate model in the Technology Library 306. For every other generic in each VHDL gate model, the initialized value (usually set to zero in model: tpd_a .VitalDelayType01 := (.000 ns, .000 ns);) is changed to an equation associated with the correlation policy.

EXPRESS MAIL LABEL NO. EL863783632US

The following shows an example of equations for referencing correlation delays for a 1X type data structure:

Given: Type 1X delay correlation on AND2_H gate VHDL

Tpd_super_rise : STRING := "1QABCDE"; (Back Annotated)

5 Tpd_super_fall : STRING := "1ABCDEF"; (Back Annotated)

Tpd_a : VitalDelayType01

:= (AND2_H_RISE(((tpd_super_rise(0)*6)+tpd_a_offset),
tpd_super_rise(1)),
AND2_H_FALL(((tpd_super_fall(0)*6)+tpd_a_offset),
tpd_super_fall(1)));

In this example, the AND2_H_RISE 2D array constant would be the name for all 2-input gate topologies to satisfy a multiple gate correlation policy. Bit 0 of tpd_super_rise is used to jump to a set of generics for a particular accessed for rise time delay resolution. A name could be aliased to a common correlation set in the 1X structure (for a 2-input topology, there are 6 generics defined). The delays for the tpd_a generic are predefined to be the 1st definition in an AND2_H correlation set, so the technology library **306** knows to use this value to select the proper set of delays. It is also known that tpd_super_rise/fall are defined to have the generics ordered the same, so bit 1 (after correlation set selector) is the tpd_a delay index to the actual delay value. The value of tpd_super_rise(1) is character "Q" which would map to the 52nd delay entry of the correlation set for tpd_a. A similar, but independent, indexing scheme is performed to get the fall time delay value.

EXPRESS MAIL LABEL NO. EL863783632US

The following shows the equation for referencing correlation delays for a 2X type data structure, using a different correlation policy:

Given: Type 2X delay correlation on common generic "tpd_b"

Tpd_super_rise : STRING := "ABCDEFQABCDE"; (Back Annotated)

5 Tpd_super_fall : STRING := "GHIJKLMNOPDEF"; (Back Annotated)

Tpd_b : VitalDelayType01

```
:= (AND2_H_RISE((tpd_super_rise(1)*tpd_b_offset),
    tpd_super_rise(7)),
    AND2_H_FALL((tpd_super_fall(1)*tpd_b_offset),
    tpd_super_fall(7)));
```

In this example the equation looks similar, but there are subtle changes that indicate the power of this mechanism using a 2X structure. The VHDL is still updated using 2D array AND2_H_RISE, but the name is now aliased to point to a common correlation structure that is based on generic names. The main difference is that the term tpd_a_offset would have a much larger value than 0 (it could be defined as the 22nd entry in the correlation set of all generics for the chip). The same 2X tpd_super_rise and tpd_super_fall values are used to designate a 2X structure of 6 independent correlation sets for 6 independent generics defined for a two input AND gate. While still using local gate level semantics for delay specification, one can take advantage of global chip level correlation optimizations.

In both examples, the two final rise and fall time values would be used to define the rise and fall times for the VHDL constant tpd_a, which would be

EXPRESS MAIL LABEL NO. EL863783632US

referenced by the VHDL model when delay values for `tpd_a` are required. Since `tpd_a` is now a constant as opposed to a generic, no back annotation of delay values is required, because these equations consist entirely of references to constants, indexed using two generics. The resolution of the actual constant values is performed once at the beginning of simulation.

5

FIG. 21 illustrates the process of compiling a chip VHDL file **316**, according to step **418**. The VHDL compiler **212** uses the chip VHDL file **316** and the technology library **306** to generate an object file **320**. The object file **320** contains the machine language executables used to simulate the chip on a particular workstation platform. The correlation VHDL file **318**, the technology library VHDL **306**, and the actual chip VHDL **316** can be compiled at separate times.

10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95

FIG. 22 illustrates performing an actual VHDL simulation. The VHDL modeling system **100** executes a VHDL simulation **420** using the reduced SDF file **314** and the object file **320** of the chip. The fact that the reduced SDF file **314** is much smaller than the original SDF file **310** increases the performance of the VHDL simulator **214** by decreasing both the run time of the simulation and the amount of memory required to contain the information. The technology library **306** binds the compiled correlation VHDL module **318** to itself at simulation time. The technology library **306** will reference the correlation array (or matrix), defined in the compiled correlation VHDL file **318**, as a VHDL package, which is an external library file in VHDL parlance, a recursive use of VHDL library function. All required files are

EXPRESS MAIL LABEL NO. EL863783632US

bound together as part of the simulation load process (all external references are resolved by the binding process, sometimes referred to as the elaboration phase of loading the simulation model). So, the chip VHDL **316** is un-altered from what would occur in a normal run. All of the delay correlations, the SDF reductions, and

5 Technology Library updates are bound to the chip VHDL **316** at simulation time.

During the simulation, the super generics are back annotated as in prior art. The equations in the updated technology library **308** now use the super generics to index into the new array structures defined in package VHDL, at step 408, to extract the actual delay values. No back annotation is required to do this - just the pointer resolution when VHDL modules are linked together in the elaboration phase of VHDL simulation (the elaboration phase is known to people versed in art of VHDL simulation models). The elimination of the back annotation step saves a considerable amount of time.

10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95
100
105
110
115
120
125
130
135
140
145
150
155
160
165
170
175
180
185
190
195
200
205
210
215
220
225
230
235
240
245
250
255
260
265
270
275
280
285
290
295
300
305
310
315
320
325
330
335
340
345
350
355
360
365
370
375
380
385
390
395
400
405
410
415
420
425
430
435
440
445
450
455
460
465
470
475
480
485
490
495
500
505
510
515
520
525
530
535
540
545
550
555
560
565
570
575
580
585
590
595
600
605
610
615
620
625
630
635
640
645
650
655
660
665
670
675
680
685
690
695
700
705
710
715
720
725
730
735
740
745
750
755
760
765
770
775
780
785
790
795
800
805
810
815
820
825
830
835
840
845
850
855
860
865
870
875
880
885
890
895
900
905
910
915
920
925
930
935
940
945
950
955
960
965
970
975
980
985
990
995
1000
1005
1010
1015
1020
1025
1030
1035
1040
1045
1050
1055
1060
1065
1070
1075
1080
1085
1090
1095
1100
1105
1110
1115
1120
1125
1130
1135
1140
1145
1150
1155
1160
1165
1170
1175
1180
1185
1190
1195
1200
1205
1210
1215
1220
1225
1230
1235
1240
1245
1250
1255
1260
1265
1270
1275
1280
1285
1290
1295
1300
1305
1310
1315
1320
1325
1330
1335
1340
1345
1350
1355
1360
1365
1370
1375
1380
1385
1390
1395
1400
1405
1410
1415
1420
1425
1430
1435
1440
1445
1450
1455
1460
1465
1470
1475
1480
1485
1490
1495
1500
1505
1510
1515
1520
1525
1530
1535
1540
1545
1550
1555
1560
1565
1570
1575
1580
1585
1590
1595
1600
1605
1610
1615
1620
1625
1630
1635
1640
1645
1650
1655
1660
1665
1670
1675
1680
1685
1690
1695
1700
1705
1710
1715
1720
1725
1730
1735
1740
1745
1750
1755
1760
1765
1770
1775
1780
1785
1790
1795
1800
1805
1810
1815
1820
1825
1830
1835
1840
1845
1850
1855
1860
1865
1870
1875
1880
1885
1890
1895
1900
1905
1910
1915
1920
1925
1930
1935
1940
1945
1950
1955
1960
1965
1970
1975
1980
1985
1990
1995
2000
2005
2010
2015
2020
2025
2030
2035
2040
2045
2050
2055
2060
2065
2070
2075
2080
2085
2090
2095
2100
2105
2110
2115
2120
2125
2130
2135
2140
2145
2150
2155
2160
2165
2170
2175
2180
2185
2190
2195
2200
2205
2210
2215
2220
2225
2230
2235
2240
2245
2250
2255
2260
2265
2270
2275
2280
2285
2290
2295
2300
2305
2310
2315
2320
2325
2330
2335
2340
2345
2350
2355
2360
2365
2370
2375
2380
2385
2390
2395
2400
2405
2410
2415
2420
2425
2430
2435
2440
2445
2450
2455
2460
2465
2470
2475
2480
2485
2490
2495
2500
2505
2510
2515
2520
2525
2530
2535
2540
2545
2550
2555
2560
2565
2570
2575
2580
2585
2590
2595
2600
2605
2610
2615
2620
2625
2630
2635
2640
2645
2650
2655
2660
2665
2670
2675
2680
2685
2690
2695
2700
2705
2710
2715
2720
2725
2730
2735
2740
2745
2750
2755
2760
2765
2770
2775
2780
2785
2790
2795
2800
2805
2810
2815
2820
2825
2830
2835
2840
2845
2850
2855
2860
2865
2870
2875
2880
2885
2890
2895
2900
2905
2910
2915
2920
2925
2930
2935
2940
2945
2950
2955
2960
2965
2970
2975
2980
2985
2990
2995
3000
3005
3010
3015
3020
3025
3030
3035
3040
3045
3050
3055
3060
3065
3070
3075
3080
3085
3090
3095
3100
3105
3110
3115
3120
3125
3130
3135
3140
3145
3150
3155
3160
3165
3170
3175
3180
3185
3190
3195
3200
3205
3210
3215
3220
3225
3230
3235
3240
3245
3250
3255
3260
3265
3270
3275
3280
3285
3290
3295
3300
3305
3310
3315
3320
3325
3330
3335
3340
3345
3350
3355
3360
3365
3370
3375
3380
3385
3390
3395
3400
3405
3410
3415
3420
3425
3430
3435
3440
3445
3450
3455
3460
3465
3470
3475
3480
3485
3490
3495
3500
3505
3510
3515
3520
3525
3530
3535
3540
3545
3550
3555
3560
3565
3570
3575
3580
3585
3590
3595
3600
3605
3610
3615
3620
3625
3630
3635
3640
3645
3650
3655
3660
3665
3670
3675
3680
3685
3690
3695
3700
3705
3710
3715
3720
3725
3730
3735
3740
3745
3750
3755
3760
3765
3770
3775
3780
3785
3790
3795
3800
3805
3810
3815
3820
3825
3830
3835
3840
3845
3850
3855
3860
3865
3870
3875
3880
3885
3890
3895
3900
3905
3910
3915
3920
3925
3930
3935
3940
3945
3950
3955
3960
3965
3970
3975
3980
3985
3990
3995
4000
4005
4010
4015
4020
4025
4030
4035
4040
4045
4050
4055
4060
4065
4070
4075
4080
4085
4090
4095
4100
4105
4110
4115
4120
4125
4130
4135
4140
4145
4150
4155
4160
4165
4170
4175
4180
4185
4190
4195
4200
4205
4210
4215
4220
4225
4230
4235
4240
4245
4250
4255
4260
4265
4270
4275
4280
4285
4290
4295
4300
4305
4310
4315
4320
4325
4330
4335
4340
4345
4350
4355
4360
4365
4370
4375
4380
4385
4390
4395
4400
4405
4410
4415
4420
4425
4430
4435
4440
4445
4450
4455
4460
4465
4470
4475
4480
4485
4490
4495
4500
4505
4510
4515
4520
4525
4530
4535
4540
4545
4550
4555
4560
4565
4570
4575
4580
4585
4590
4595
4600
4605
4610
4615
4620
4625
4630
4635
4640
4645
4650
4655
4660
4665
4670
4675
4680
4685
4690
4695
4700
4705
4710
4715
4720
4725
4730
4735
4740
4745
4750
4755
4760
4765
4770
4775
4780
4785
4790
4795
4800
4805
4810
4815
4820
4825
4830
4835
4840
4845
4850
4855
4860
4865
4870
4875
4880
4885
4890
4895
4900
4905
4910
4915
4920
4925
4930
4935
4940
4945
4950
4955
4960
4965
4970
4975
4980
4985
4990
4995
5000
5005
5010
5015
5020
5025
5030
5035
5040
5045
5050
5055
5060
5065
5070
5075
5080
5085
5090
5095
5100
5105
5110
5115
5120
5125
5130
5135
5140
5145
5150
5155
5160
5165
5170
5175
5180
5185
5190
5195
5200
5205
5210
5215
5220
5225
5230
5235
5240
5245
5250
5255
5260
5265
5270
5275
5280
5285
5290
5295
5300
5305
5310
5315
5320
5325
5330
5335
5340
5345
5350
5355
5360
5365
5370
5375
5380
5385
5390
5395
5400
5405
5410
5415
5420
5425
5430
5435
5440
5445
5450
5455
5460
5465
5470
5475
5480
5485
5490
5495
5500
5505
5510
5515
5520
5525
5530
5535
5540
5545
5550
5555
5560
5565
5570
5575
5580
5585
5590
5595
5600
5605
5610
5615
5620
5625
5630
5635
5640
5645
5650
5655
5660
5665
5670
5675
5680
5685
5690
5695
5700
5705
5710
5715
5720
5725
5730
5735
5740
5745
5750
5755
5760
5765
5770
5775
5780
5785
5790
5795
5800
5805
5810
5815
5820
5825
5830
5835
5840
5845
5850
5855
5860
5865
5870
5875
5880
5885
5890
5895
5900
5905
5910
5915
5920
5925
5930
5935
5940
5945
5950
5955
5960
5965
5970
5975
5980
5985
5990
5995
6000
6005
6010
6015
6020
6025
6030
6035
6040
6045
6050
6055
6060
6065
6070
6075
6080
6085
6090
6095
6100
6105
6110
6115
6120
6125
6130
6135
6140
6145
6150
6155
6160
6165
6170
6175
6180
6185
6190
6195
6200
6205
6210
6215
6220
6225
6230
6235
6240
6245
6250
6255
6260
6265
6270
6275
6280
6285
6290
6295
6300
6305
6310
6315
6320
6325
6330
6335
6340
6345
6350
6355
6360
6365
6370
6375
6380
6385
6390
6395
6400
6405
6410
6415
6420
6425
6430
6435
6440
6445
6450
6455
6460
6465
6470
6475
6480
6485
6490
6495
6500
6505
6510
6515
6520
6525
6530
6535
6540
6545
6550
6555
6560
6565
6570
6575
6580
6585
6590
6595
6600
6605
6610
6615
6620
6625
6630
6635
6640
6645
6650
6655
6660
6665
6670
6675
6680
6685
6690
6695
6700
6705
6710
6715
6720
6725
6730
6735
6740
6745
6750
6755
6760
6765
6770
6775
6780
6785
6790
6795
6800
6805
6810
6815
6820
6825
6830
6835
6840
6845
6850
6855
6860
6865
6870
6875
6880
6885
6890
6895
6900
6905
6910
6915
6920
6925
6930
6935
6940
6945
6950
6955
6960
6965
6970
6975
6980
6985
6990
6995
7000
7005
7010
7015
7020
7025
7030
7035
7040
7045
7050
7055
7060
7065
7070
7075
7080
7085
7090
7095
7100
7105
7110
7115
7120
7125
7130
7135
7140
7145
7150
7155
7160
7165
7170
7175
7180
7185
7190
7195
7200
7205
7210
7215
7220
7225
7230
7235
7240
7245
7250
7255
7260
7265
7270
7275
7280
7285
7290
7295
7300
7305
7310
7315
7320
7325
7330
7335
7340
7345
7350
7355
7360
7365
7370
7375
7380
7385
7390
7395
7400
7405
7410
7415
7420
7425
7430
7435
7440
7445
7450
7455
7460
7465
7470
7475
7480
7485
7490
7495
7500
7505
7510
7515
7520
7525
7530
7535
7540
7545
7550
7555
7560
7565
7570
7575
7580
7585
7590
7595
7600
7605
7610
7615
7620
7625
7630
7635
7640
7645
7650
7655
7660
7665
7670
7675
7680
7685
7690
7695
7700
7705
7710
7715
7720
7725
7730
7735
7740
7745
7750
7755
7760
7765
7770
7775
7780
7785
7790
7795
7800
7805
7810
7815
7820
7825
7830
7835
7840
7845
7850
7855
7860
7865
7870
7875
7880
7885
7890
7895
7900
7905
7910
7915
7920
7925
7930
7935
7940
7945
7950
7955
7960
7965
7970
7975
7980
7985
7990
7995
8000
8005
8010
8015
8020
8025
8030
8035
8040
8045
8050
8055
8060
8065
8070
8075
8080
8085
8090
8095
8100
8105
8110
8115
8120
8125
8130
8135
8140
8145
8150
8155
8160
8165
8170
8175
8180
8185
8190
8195
8200
8205
8210
8215
8220
8225
8230
8235
8240
8245
8250
8255
8260
8265
8270
8275
8280
8285
8290
8295
8300
8305
8310
8315
8320
8325
8330
8335
8340
8345
8350
8355
8360
8365
8370
8375
8380
8385
8390
8395
8400
8405
8410
8415
8420
8425
8430
8435
8440
8445
8450
8455
8460
8465
8470
8475
8480
8485
8490
8495
8500
8505
8510
8515
8520
8525
8530
8535
8540
8545
8550
8555
8560
8565
8570
8575
8580
8585
8590
8595
8600
8605
8610
8615
8620
8625
8630
8635
8640
8645
8650
8655
8660
8665
8670
8675
8680
8685
8690
8695
8700
8705
8710
8715
8720
8725
8730
8735
8740
8745
8750
8755
8760
8765
8770
8775
8780
8785
8790
8795
8800
8805
8810
8815
8820
8825
8830
8835
8840
8845
8850
8855
8860
8865
8870
8875
8880
8885
8890
8895
8900
8905
8910
8915
8920
8925
8930
8935
8940
8945
8950
8955
8960
8965
8970
8975
8980
8985
8990
8995
9000
9005
9010
9015
9020
9025
9030
9035
9040
9045
9050
9055
9060
9065
9070
9075
9080
9085
9090
9095
9100
9105
9110
9115
9120
9125
9130
9135
9140
9145
9150
9155
9160
9165
9170
9175
9180
9185
9190
9195
9200
9205
9210
9215
9220
9225
9230
9235
9240
9245
9250
9255
9260
9265
9270
9275
9280
9285
9290
9295
9300
9305
9310
9315
9320
9325
9330
9335
9340
9345

EXPRESS MAIL LABEL NO. EL863783632US

or in a distributed fashion where different elements are spread across several interconnected computer systems. Any kind of computer system - or other apparatus adapted for carrying out the methods described herein - is suited. A typical combination of hardware and software could be a general-purpose computer system with a computer program that, when being loaded and executed, controls the computer system such that it carries out the methods described herein.

The present invention can also be embedded in a computer program product, which comprises all the features enabling the implementation of the methods described herein, and which - when loaded in a computer system - is able to carry out these methods. Computer program means or computer program in the present context mean any expression, in any language, code or notation, of a set of instructions intended to cause a system having an information processing capability to perform a particular function either directly or after either or both of the following

a) conversion to another language, code or, notation; and b) reproduction in a different material form.

A computer system may include, inter alia, one or more computers and at least a computer readable medium, allowing a computer system, to read data, instructions, messages or message packets, and other computer readable information from the computer readable medium. The computer readable medium may include non-volatile memory, such as ROM, Flash memory, Disk drive memory, CD-ROM, and other permanent storage. Additionally, a computer readable medium

EXPRESS MAIL LABEL NO. EL863783632US

may include, for example, volatile storage such as RAM, buffers, cache memory, and network circuits. Furthermore, the computer readable medium may comprise computer readable information in a transitory state medium such as a network link and/or a network interface, including a wired network or a wireless network, that 5 allow a computer system to read such computer readable information.

Although specific embodiments of the invention have been disclosed, those having ordinary skill in the art will understand that changes can be made to the specific embodiments without departing from the spirit and scope of the invention. 10 The scope of the invention is not to be restricted, therefore, to the specific embodiments, and it is intended that the appended claims cover any and all such applications, modifications, and embodiments within the scope of the present invention.

15 What is claimed is: